Eliminating Redundant Association Rules in Multi-level Datasets

Gavin Shaw, Yue Xu and Shlomo Geva

Faculty of Information Technology, Queensland University of Technology, Brisbane, Australia

Abstract — *Association rule mining plays an important job in knowledge and information discovery and there are many approaches available. However, there are still shortcomings with the quality of the discovered rules. Often the number of the discovered rules is huge and many of them are redundant, especially in the case of multi-level datasets. Previous work has shown that the mining of non-redundant rules is a promising approach to solving this problem. However, work by Pasquier et. al. [14] and Xu & Li [17,18] is only focused on single level datasets. In this paper, we propose an extension to this previous work that allows them to remove hierarchically redundant rules from multi-level datasets. We also show that the resulting concise representation of non-redundant association rules is lossless since all association rules can be derived from the representation. Experiments show that our extension can effectively generate multilevel non-redundant rules.*

Keywords: redundant association rules, multi-level datasets

1. INTRODUCTION

Since its introduction in [1], association rule mining has become both an important and widely used data mining technique. The aim of this technique is to extract frequent patterns, interesting co-occurrences and associations amongst sets of items in large transactional databases. Traditionally there are two steps in obtaining association rules: firstly, determining the frequent patterns or itemsets using the constraint of minimal support and secondly generating the rules from these frequent patterns/itemsets using the constraint of minimal confidence. With this approach, the basis of an interesting or useful rule is that its confidence exceeds a user defined threshold. This approach is widely known as the frequent itemset approach. Much work has been done in developing more and more efficient algorithms or data structures to make computing these rules quicker. Much effort has been focused on improving the determination of the frequent itemsets [2,3,4,7,15].

Another technique that has developed from the traditional frequent itemset approach is the use of frequent closed itemsets, which has originated from the mathematical theory of Formal Concept Analysis (FCA). It was shown to be a powerful technique for data analysis [13,19]. Its major advantage is its ability to reduce the number of rules as well as provide a more concise representation which is lossless. Usually too many association rules containing redundancies are discovered; often too many to comprehend. Using frequent closed itemsets the issue of redundancy can be dealt with by deriving non-redundant association rules [14,17,18,20]. However, this work has only dealt with redundancy in single level datasets. Multi-level datasets (in which the items are not all at the same concept level) contain information at different levels. The approaches used to find frequent itemsets in single level datasets miss information, as they only look at one level in the dataset. Thus techniques that consider all the levels are needed [6,8,9,11,12]. However, rules derived from multi-level datasets can have the same issues with redundancy as those from a single level dataset. While approaches used to remove redundancy in single level datasets [14,18] can be adapted for use in multi-level datasets, they still fail to remove all of the redundancies, namely the redundancy of hierarchy, where one rule at a given level gives the same information as another rule at a different level.

This paper looks into this hierarchical redundancy and proposes an approach from which more concise non-redundant rules can be derived. We use the same definition for non-redundant rules, in which minimal antecedent and maximal consequents are desired, as defined by Xu & Li [17,18]. But to the definition we add a requirement that considers the level of the item(s) in the rule in determining redundancy. By doing so, more redundant association rules can be eliminated. We also show that it is possible to derive all of the association rules from this more concise set of basis rules and thus there is no loss of information in this basis set.

The paper is organized as follows. Section 2 discusses related work. The basics behind association rule mining are given in Section 3. We present the definition of hierarchical redundancy and introduce our approach for deriving the non-redundant exact basis rule set and how to recover all the exact rules in Section 4. Experiments and results are presented in Section 5. Lastly, Section 6 concludes the paper and suggests directions for possible future work.

2. RELATED WORK

Much work in the field of association rule mining has focused on finding more and more efficient ways to discover all of the rules. This has meant less work has focused on the issue of the quality of the discovered association rules. Furthermore, complete rule enumeration is very often intractable in data sets with a very large number of multi-valued attributes.

One approach that has been argued is that it is not the number of rules that overwhelm a person, but the lack of organization and presentation to make them easier to analyze [10,11]. Here, rules are grouped into sets and generalized, thus redundant rules can remain.

The approach being taken is to determine which rules are redundant and remove them, thus reducing the number of rules a user has to deal with while not reducing the information content [14,18,20]. These approaches are showing a lot of promise and indeed work done in [18] shows that reductions of over 80% can be achieved. This kind of work has only focused on datasets where all items are at the same concept level. Thus they do not need to worry about or consider redundancy that can occur when there is a hierarchy among items.

A multi-level dataset is one which has an implicit taxonomy or concept tree, like shown in Figure 1. The items in the dataset exist at the lowest concept level but are part of a hierarchical structure and organization. Thus for example, 'Old Mills' is an item at the lowest level of the taxonomy but it also belongs to the higher concept category of 'bread' and also the more refined category 'white bread'.

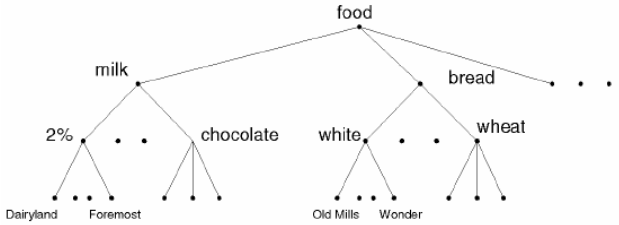

Figure 1. The taxonomy of a multi-level dataset.

Because of the hierarchical nature of a multi-level dataset, a new approach to finding frequent itemsets for multi-level datasets has to be considered. Work has been done in adapting approaches originally made for single level datasets into techniques usable on multi-level datasets. Work presented in [5] shows one of the earliest approaches proposed to find frequent itemsets in multi-level datasets and later was revisited in [6]. This work primarily focused on finding frequent itemsets at each of the levels in the dataset and did not focus heavily on cross-level itemsets (those itemsets that are composed of items from two or more different levels). Referring to Figure 1 for an example, the frequent itemset {'Dairyland-2%-milk', 'white-bread'} is a cross-level itemset as the first item is from the lowest level, while the second item is from a different concept level, namely the next level up. In fact the cross-level ideas were an addition to the work being proposed. Further work proposed an approach which included finding cross-level frequent itemsets [16]. This later work also performs more pruning of the dataset to make finding the frequent itemsets more efficient.

However, even with all this work the focus has been on finding the frequent itemsets as efficiently as possible and the issue of quality and/or redundancy in single level datasets. Some brief work presented by Han & Fu [5] discusses removing rules which are hierarchically redundant, but it relies on the user giving an expected confidence variation margin to determine redundancy. There appears to be a void in dealing with hierarchical redundancy in association rules derived from multi-level datasets. This work attempts to fill that void and show an approach to deal with hierarchical redundancy without losing any information.

3. MINING FREQUENT PATTERNS

From the beginning of association rule mining in [1], the first step has always been to find the frequent patterns or itemsets. The simplest way to do this is through the use of the Apriori algorithm [2]. However, Apriori is not designed to work on extracting frequent itemsets at multiple levels in a multi-level dataset. It is designed for use on single level datasets. But, it has been adapted for multi-level datasets.

One adaptation of Apriori to multi-level datasets is the ML_T2L1 algorithm [5,6]. The ML_T2L1 algorithm uses a transaction table that has the hierarchy information encoded into it. Each level in the dataset is processed individually. Firstly, level 1 (the highest level in the hierarchy) is analysed for large 1-itemsets using Apriori. The list of level 1 large 1-itemsets is then used to filter and prune the transaction dataset of any item that does not have an ancestor in the level 1 large 1-itemset list and remove any transaction which has no frequent items (thus contains only infrequent items when assessed using the level 1 large 1-itemset list). From the level 1 large 1-itemset list, level 1 large 2-itemsets are derived (using the filter dataset). Then level 1 large 3-itemsets are derived and so on, until there are no more frequent itemsets to discover at level 1. Since ML_T2L1 defines that only the items that are descendant from frequent items at level 1 (essentially they must descend from level 1 large 1-itemsets) can be frequent themselves, the level 2 itemsets are derived from the filtered transaction table. For level 2, the large 1-itemsets are discovered, from which the large 2-itemsets are derived and then large 3-itemsets etc. After all the frequent itemsets are discovered at level 2, the level 3 large 1-itemsets are discovered (from the same filtered dataset) and so on. ML_T2L1 repeats until either all levels are searched using Apriori or no large 1-itemsets are a found at a level.

As the original work shows [5,6], ML_T2L1 does not find cross-level frequent itemsets. We have added the ability for it to do this. At each level below 1 (so starting at level 2) when large 2-itemsets or later are derived the Apriori algorithm is not restricted to just using the large n-1-itemsets at the current level, but can generate combinations using the large itemsets from higher levels. The only restrictions on this are that the derived frequent itemset(s) can not contain an item that has an ancestor-descendant relationship with another item within the same itemset and that the minimum support threshold used is that of the current level being

processed (which is actually the lowest level in the itemset).

A second, more recent adaptation of Apriori for use in multi-level datasets is a top-down progressive deepening method by Thakur, Jain & Paradasani in [16]. This approach was developed to find level-crossing association rules by extending existing multi-level mining techniques and uses reduced support and refinement of the transaction table at every hierarchy level. This algorithm works very similarly to ML_T2L1 presented previously in that it uses a transaction table which has the hierarchy encoded into it and each level is processed individually, one at a time. Initially, level 1 is processed, followed by level 2, 3 and so on until the lowest level is reached and processed, or a level generates no large 1-itemsets. At each level, the large 1-itemsets are first derived and are then used to filter / prune the transaction table (as described for ML_T2L1). This filtering happens at every level, not just level 1, like in ML_T2L1. Then large 2-itemsets, 3-itemsets etc are derived from the filtered table. When it comes to level 2 and lower, the itemsets are not restricted to just the current level, but can include itemsets from large itemset lists of higher levels. This is how level-crossing association rules will be found. For the itemsets that span multiple levels, the minimum support threshold of the lowest level in the itemset is used as the threshold to determine whether the itemset is frequent / large.

The two algorithms mentioned above have been used to generate frequent itemsets in our experiments which are explained in Section 5.

4. GENERATION OF NON-REDUNDANT MULTI-LEVEL ASSOCIATION RULES

The use of frequent itemsets as the basis for association rule mining often results in the generation of a large number of rules. This is a widely recognized problem. More recent work has demonstrated that the use of closed itemsets and generators can reduce the number of rules generated [14,17,18,20]. This has helped to greatly reduce redundancy in the rules derived from single level datasets. Despite this, redundancy still exists in the rules generated from multi-level datasets even when using some of the methods designed to remove redundancy. This redundancy we call hierarchical redundancy. Here in this section we first introduce hierarchical redundancy in multi-level datasets and then we detail our work to remove this redundancy without losing information.

4.1 Hierarchical Redundancy

Whether a rule is interesting and/or useful is usually determined through the support and confidence values that it has. However, this does not guarantee that all of the rules that have a high enough support and confidence actually convey new information. To demonstrate this, the following is an example transaction table for a multi-level dataset (Table 1).

Table 1. Simple multi-level transaction dataset.

Transaction ID	Items
1	[1-1-1, 1-2-1, 2-1-1, 2-2-1]
2	[1-1-1, 2-1-1, 2-2-2, 3-2-3]
3	[1-1-2, 1-2-2, 2-2-1, 4-1-1]
4	[1-1-1, 1-2-1]
5	[1-1-1, 1-2-2, 2-1-1, 2-2-1, 4-1-3]
6	[1-1-3, 3-2-3, 5-2-4]
7	[1-3-1, 2-3-1]
8	[3-2-3, 4-1-1, 5-2-4, 7-1-3]

Table 2. Frequent itemsets.

1-itemsets	2-itemsets	3-itemsets
[1-*-*]	[1-*-*, 2-*-*]	[1-*-*, 2-1-*, 2-2-*]
[2-*-*]	[1-*-*, 2-1-*]	[2-*-*, 1-1-*, 1-2-*]
[1-1-*]	[1-*-*, 2-2-*]	[1-1-*, 1-2-*, 2-2-*]
[1-2-*]	[2-*-*, 1-1-*]	[1-1-*, 2-1-*, 2-2-*]
[2-1-*]	[2-*-*, 1-2-*]	[1-*-*, 2-1-*, 2-2-*]
[2-2-*]	[1-1-*, 1-2-*]	[1-1-*, 2-1-1, 2-2-*]
[1-1-1]	[1-1-*, 2-1-*]	[1-1-*, 2-2-1, 1-2-*]
[2-1-1]	[1-1-*, 2-2-*]	[2-1-*, 1-1-1, 2-2-*]
[2-2-1]	[1-2-*, 2-2-*]	[2-2-*, 1-1-1, 2-1-1]
	[2-1-*, 2-2-*]	
	[1-*-*, 2-1-1]	
	[1-*-*, 2-2-1]	
	[2-*-*, 1-1-1]	
	[1-1-*, 2-1-1]	
	[1-1-*, 2-2-1]	
	[1-2-*, 1-1-1]	
	[1-2-*, 2-2-1]	
	[2-1-*, 1-1-1]	
	[2-2-*, 1-1-1]	
	[2-2-*, 2-1-1]	
	[1-1-1, 2-1-1]	

This simple multi-level dataset has 3 levels with each item belonging to the lowest level. The item ID in the table store/holds the hierarchy information for each item. Thus the item 1-2-1 belongs to the first category at level 1 and for level 2 it belongs to the second sub-category of the first level 1 category. Finally at level 3 it belongs to the first sub-category of the parent category at level 2. From this transaction set we use the ML_T2L1 algorithm with the cross level add-on (as described previously) and a minimum support value of 4 for level 1 and 3 for levels 2 and 3. The following frequent itemsets are discovered (Table 2). From these frequent itemsets the closed itemsets and generators are derived (Table 3). The itemsets, closed itemsets and generators come from all three levels.

Finally from the closed itemsets and generators the association rules can be generated. In this example we use the ReliableExactRule approach presented in [17,18] to generate the exact basis rules. The discovered rules are from multiple levels and include cross-level rules (due to cross-level frequent itemsets). The ReliableExactRule approach can remove redundant rules, but as we will show, it does not remove hierarchy redundancy. The rules given in Table 4

are derived from the closed itemsets and generators in Table 3 when the minimum confidence threshold is set to 0.5 or 50% (Table 4).

Table 3. Frequent closed itemsets and generators derived from the frequent itemsets in Table 2.

Closed Itemsets	Generators
[1-*-*]	[1-*-*]
[1-1-*]	[1-1-*]
[1-1-1]	[1-1-1]
[1-*-*, 2-2-*]	[2-2-*]
[2-*-*, 1-1-*]	[2-*-*, 1-1-*]
[1-1-*, 1-2-*]	[1-2-*]
[1-1-*, 2-2-*]	[2-2-*]
[1-*-*, 2-2-1]	[2-2-1]
[2-*-*, 1-1-1]	[2-*-*, 1-1-1]
[1-2-*, 1-1-1]	[1-2-*, 1-1-1]
[1-*-*, 2-1-*, 2-2-*]	[2-1-*]
[2-*, 1-1-*, 1-2-*]	[2-*-*, 1-2-*]
[1-1-*, 1-2-*, 2-2-*]	[1-2-*, 2-2-*]
[1-1-*, 2-1-*, 2-2-*]	[2-1-*]
[1-*-*, 2-1-1, 2-2-*]	[2-1-1]
[1-1-*, 2-1-1, 2-2-*]	[2-1-1]
[1-1-*, 2-2-1, 1-2-*]	[2-2-1]
[2-1-*, 1-1-1, 2-2-*]	[2-1-*] [2-2-*, 1-1-1]
[2-2-*, 1-1-1, 2-1-1]	[2-1-1] [2-2-*, 1-1-1]

Table 4. Exact basis association rules derived from closed itemsets and generators in Table 3.

No.	Rule	Supp
1	[2-2-*] ==> [1-*-*]	0.571
2	[1-2-*] ==> [1-1-*]	0.571
3	[2-2-*] ==> [1-1-*]	0.571
4	[2-2-1] ==> [1-*-*]	0.428
5	[2-1-*] ==> [1-*-*, 2-2-*]	0.428
6	[2-1-*] ==> [1-1-*, 2-2-*]	0.428
7	[2-1-1] ==> [1-*-*, 2-2-*]	0.428
8	[2-1-1] ==> [1-1-*, 2-2-*]	0.428
9	[2-2-1] ==> [1-1-*, 1-2-*]	0.428
10	[2-1-*] ==> [1-1-1, 2-2-*]	0.428
11	[2-2-*, 1-1-1] ==> [2-1-*]	0.428
12	[2-1-1] ==> [2-2-*, 1-1-1]	0.428
13	[2-2-*, 1-1-1] ==> [2-1-1]	0.428

The ReliableExactRule algorithm lists all of the rules (in Table 4) as important and non-redundant. However, we argue that there are still redundant rules. This type of redundancy is beyond what the ReliableExactRule algorithm was designed for. Looking at the rules in Table 4 we claim that rule 4 is redundant to rule 1, rule 7 is redundant to rule 5, rule 8 is redundant to rule 6 and rule 12 is redundant to rule 10. For example, the item 2-2-1 (from rule 4) is a child of the more general/abstract item 2-2-* (from rule 1). Thus rule 4 is in fact a more specific version of rule 1. Because we know that rule 1 says 2-2-* is enough to fire the rule with consequent C, whereas rule 4 requires 2-2-1 to fire with consequent C, any item that is a descendant of 2-2-* will cause a rule to fire with consequent C. It does not have to be 2-2-1. Thus rule 4 is more restrictive. Because 2-2-1 is part of 2-2-* having rule 4 does not actually bring any new information to the user, as the information contained in it is actually part of the information contained in rule 1. Thus rule 4 is redundant. We define hierarchical redundancy in exact association rules through the following definition.

Definition 1: Let $R_1 = X_1 => Y$ and $R_2 = X_2 => Y$ be two exact association rules, with exactly the same itemset Y as the consequent. Rule R_1 is redundant to rule R_2 if (1) the itemset X_1 is made up of items where at least one item in X_1 is descendant from the items in X_2 and (2) the itemset X_2 is entirely made up of items where at least one item in X_2 is an ancestor of the items in X_1 and (3) the other non-ancestor items in X_2 are all present in itemset X_1.

From this definition, if for an exact association rule $X_1 => Y_1$ there does not exist any other rule $X_2 => Y_2$ such that at least one item in X_1 shares an ancestor-descendant relationship with X_2 containing the ancestor(s) and all other items X_2 are present in X_1, then $X_1 => Y_1$ is a non-redundant rule. To test for redundancy, we take this definition and add another condition for a rule to be considered valid. A rule X => Y is valid if it has no ancestor-descendant relationship between any items in itemsets X and Y. Thus for example 1-2-1 => 1-2-* is not a valid rule, but 1-2-1 => 1-1-3 is a valid rule. If this condition is not met by any rule $X_2 => Y_2$ when testing to see if $X_1 => Y_1$ is redundant to $X_2 => Y_2$, then $X_1 => Y_1$ is a non-redundant rule as $X_2 => Y_2$ is not a valid rule.

4.2 Generating Exact Basis Rules

As previous work has shown [14,17,18] using frequent closed itemsets in the generation of association rules can reduce the quantity of discovered rules. Because we wish to remove redundancy on top of the redundancy already being removed, our approach uses the closed itemsets and generators to discover the non-redundant rules. Pasquier et. al. [14] and Xu & Li [17,18] have both proposed condensed/more concise bases to represent non-redundant exact rules. Exact rules refer to rules whose confidence is 1. The proposed approach will be extended to other rules (i.e., so called approximate rules). The following definitions outline these two bases:

Definition 2: For the Min-MaxExact (MME) basis, C is the set of the discovered frequent closed itemsets. For each closed itemset c in C, G_c is the set of generators for c. From this the exact basis for min-max is:

$$\text{MME} = \{r : g => (c \setminus g) \mid c \in C \,\&\, g \in G_c \,\&\, g \neq c\}$$

Definition 3: For the ReliableExactRule (RER) basis, C is the set of the discovered frequent closed itemsets. For each closed itemset c in C, G_c is the set of generators for c. Thus the exact basis for reliable exact is:

$$\text{RER} = \left\{ \begin{array}{l} r : g => (c \setminus g) \mid c \in C \,\&\, g \in G_c \,\& \\ \neg(g \supseteq ((c \setminus c') \cup g')) \\ where, c' \in C' \& c' \subset c \,\&\, g' \in G_{c'} \end{array} \right\}$$

To each of the two definitions we add our definition (1) for generating the non-redundant multi-level association

rules. Thus our modified approaches to deriving the exact basis rules are as follows:

Definition 4: For the Min-MaxExact basis with HRR (MME-HRR), C is the set of the discovered frequent closed itemsets. For each closed itemset c in C, G_c is the set of generators for c. Also, G is the set of all generators in which g' is a generator from which the closed itemset c' is the closed itemset from the set of closed itemsets C derived from g' ($C_{g'}$). From this the exact basis for min-max is now:

$= \{r : g \Rightarrow (c \setminus g) \mid c \in C \,\&\, g \in G_c \,\&\, g \neq c \,\&$

there exists no

$r' : g' \Rightarrow (c' \setminus g') \mid g' \in G \,\&\, g \neq g' \,\&\, c' \in C_{g'} \,\&\, g' \neq c'$

where g is descendant set of g' and g' is ancestor set of g and $(c \setminus g) = (c' \setminus g')$ and g' has no ancestors or descendants of $(c' \setminus g')\}$

Definition 5: For the ReliableExactRule basis with HRR (RER-HRR), C is the set of the discovered frequent closed itemsets. For each closed itemset c in C, G_c is the set of generators for c. Also, G is the set of all generators in which g_1 is a generator from which the closed itemset c_1 is the closed itemset from the set of closed itemsets C derived from g_1 (G_{g_1}). Thus the exact basis for reliable exact is now:

$= \{r : g \Rightarrow (c \setminus g) \mid c \in C \,\&\, g \in G_c \,\&$

$\neg(g \supseteq ((c \setminus c') \cup g')) \,\&$

there exists no

$r_1 : g_1 \Rightarrow (c_1 \setminus g_1) \mid g_1 \in G \,\&\, g \neq g_1 \,\&\, c_1 \in C_{g_1}$

$\&\, g_1 \neq c_1$

where g is descendant set of g_1 and g_1 is ancestor set of g and $(c \setminus g) = (c_1 \setminus g_1)$ and g_1 has no ancestors or descendants of $(c_1 \setminus g_1), where, c' \in C' \,\&\, c' \subset c \,\&\, g' \in G_{c'}\}$

Thus the algorithms to extract non-redundant multi-level rules using either Min-MaxExact or ReliableExactRule algorithms as the foundation are given as follows:

Algorithm 1: Min-MaxExact with HRR
Input: Set of frequent closed itemsets and generators
Output: Set of non-redundant multi-level rules
1. MinMaxExact ← Ø
2. for k = 1 to v do
3. for all k-generator $g \in FC_k$ do
4. nonRedundant = true
5. if $(g \neq g.closure)$
6. for all $g' \in G$
7. if $(g' \neq g)$
8. if (g' ancestor of g) & (g descendant of g') & (c' = c) & !(g' ancestor of (c' \ g')) & !(g' descendant of (c' \ g'))
9. nonRedundant = false
10. break
11. if nonRedundant
12. insert {r : g => (c \ g), g.supp} in MinMaxExact
13. return MinMaxExact

Algorithm 2: ReliableExactRule with HRR
Input: Set of frequent closed itemsets and generators
Output: Set of non-redundant multi-level rules
1. exactRules ← Ø
2. for all $c \in C$
3. for all $g \in G_c$
4. nonRedundant = false
5. if $\forall c' \in C$ such that $c' \subset c \,\&\, \forall g' \in G_{c'}$ we have $\neg(g \supseteq ((c \setminus c') \cup g'))$
6. nonRedundant = true
7. else
8. nonRedundant = false
9. break
10. for all $g_1 \in G$
11. if $g_1 \neq g$
12. if (g' ancestor of g) & (g descendant of g') & (c' = c) & !(g' ancestor of (c' \ g')) & !(g' descendant of (c' \ g'))
13. nonRedundant = false
14. break
15. if nonRedundant
16. insert {r : g => (c \ g), g.supp} in exactRules
17. return exactRules

The complexity of the original MinMaxExact is O(n), where n is the number of generators derived from the frequent itemsets. For the algorithm Min-MaxExact with HRR, before generating a rule, we need to scan all generators to determine whether it is hierarchically redundant. Therefore, the complexity of the algorithm Min-MaxExact with HRR is O(n^2). For the original ReliableExactRule algorithm, the complexity is O(n^2). Our modified algorithm ReliableExactRule with HRR, does not change its complexity, i.e., O(n^2). For large datasets, with the O(n^2) complexity, the two proposed methods may have efficiency problems. This issue will be addressed in our future work.

5. EXPERIMENTAL RESULTS

Experiments were conducted to test and evaluate the effectiveness of the proposed hierarchically non-redundant exact basis and to confirm that it is also a lossless basis set. This section presents and details the experiments and their results.

5.1 Datasets

We used 7 datasets to test our approach to discover whether it reduced the size of the exact basis rule set and to test that the basis set was lossless, meaning all the rules could be recovered. We used the same datasets used by Han & Fu [5,6] and Thakur, Jain & Paradasani [16] which had seven and eight transactions respectively and are named H1 and T1 respectively. We also used 7 randomly built datasets which were composed of 10, 20, 50, 200 and 500 transactions. The key statistics for these built datasets are detailed in Table 5. We were limited to small datasets due to

efficiency problems suffered by the algorithms used to find the frequent itemsets. Because our focus was on developing a new non-redundant association rule mining algorithm we did not devote effort into developing a more efficient method for discovering frequent itemsets. Developing a more efficient frequent itemset finding algorithm will be part of our future work.

Table 5. Dataset statistics.

Dataset Parameters	T2	T3	T4	T5	T6
No. of transactions	10	20	50	200	500
Average no. of items per transaction	5	7	7	7	20
No. of items on the top concept level	5	10	10	10	10
No. of levels in the hierarchy	3	4	4	4	4
Average no. of child items a given item has (except items on the lowest concept level)	3	4	4	4	4

The experiments aim to find associations among the items in each of the datasets. The process to discover the association rules involves three steps. Firstly, the frequent itemsets are discovered through the use of minimal support values for each hierarchy level. We have implemented two approaches to find the frequent itemsets; Han & Fu's ML_T2L1 approach presented in [5,6] with the addition to the base algorithm so as to find cross-level itemsets, and Thakur, Jain & Paradasani's algorithm (referred to as CLI) to find cross-level itemsets (along with normal itemsets) presented in [16]. Second, from the frequent itemsets, the frequent closed itemsets and generators are derived. We have implemented the CLOSE+ algorithm proposed by Pasquier et. al. in [14] to achieve this. Finally, the association rules are built. In these experiments we derive the rules using Pasquier's et. al. Min-MaxExact (referred to as MME) [14], Xu & Li's ReliableExactRule approach (referred to as RER) [17,18], a modified version of Pasquier's et. al. work in [14] to include removing hierarchical redundancy (referred to as MME with HRR) and a modified version of Xu & Li's work in [17,18] to include removing hierarchical redundancy (referred to as RER with HRR).

5.2 Results

The primary objective of the experiments is to determine how well our proposed work performs at removing / reducing hierarchical redundancy in datasets even when other redundancy eliminating processes are included. The other objective is to ensure and demonstrate that this approach is lossless and no information is lost. We have defined our approach earlier in Section 4.B to remove redundant rules in multi-level datasets and thus the exact basis should be smaller in size when it is utilized. We also confirm that our approach can recover all exact rules from multi-level datasets by comparing the modified versions of Min-MaxExact and ReliableExactRule (which include our work to remove hierarchically redundant rules) against unmodified versions for each dataset to ensure that each recover the same set of exact rules. We also compare the size of the exact basis set generated by each of the four approaches to see what reduction in the basis set can be achieved. For all of the testing undertaken, the minimum confidence threshold for the association rules was set at 0.5. Tables 6, 7, 8 & 9 present the results obtained from each of the datasets showing the percentage reduction achieved.

As can be seen, the use of our approach reduces the exact basis rule set for all cases we tested. In some instances the basis set was only reduced by a few rules, but in other cases there was a more significant reduction in the size of the basis set. For example, in Table 8 for dataset T4 there was a reduction of 148 rules from 577 to 429, which is about 25.5%, and in Table 6, the reduction was around 46 to 47% for dataset H1 and nearly 36% for dataset T2. By using this approach we have successfully reduced the size of the exact basis and by doing so it may help to make it more possible to effectively use the extracted association rules without overwhelming a user.

Table 6. Results for datasets with three hierarchy levels where ML_T2L1 with cross level add-on is used to extract frequent itemsets.

	Exact Basis						Exact Rules
Data set	MME	MME with HRR	%	RER	RER with HRR	%	
H1	21	11	47	15	8	46	25
T1	15	10	33	13	9	31	39
T2	106	68	36	80	58	27	976

Table 7. Results for datasets with three hierarchy levels where CLI is used to extract frequent itemsets.

	Exact Basis						Exact Rules
Data set	MME	MME with HRR	%	RER	RER with HRR	%	
H1	9	7	22	5	4	20	9
T1	2	1	50	2	1	50	2
T2	62	42	32	46	33	28	299

Table 8. Results for datasets with four hierarchy levels where ML_T2L1 with cross level add-on is used to extract frequent itemsets.

	Exact Basis						Exact Rules
Data set	MME	MME with HRR	%	RER	RER with HRR	%	
T3	174	134	23	113	89	21	736
T4	577	429	25	383	305	20	1584
T5	450	405	10	315	287	9	759
T6	725	602	17	91	80	12	725

Table 9. Results for datasets with four hierarchy levels where CLI is used to extract frequent itemsets.

	Exact Basis						Exact
Data set	MME	MME with HRR	%	RER	RER with HRR	%	Rules
T3	44	39	11	29	26	10	90
T4	356	271	24	244	196	19	666
T5	180	174	3	121	116	4	212
T6	325	293	10	53	47	11	325

For each test conducted we also checked the expanded exact association rules, i.e., to derive all exact rules from the exact basis. The tables show the number of expanded rules for each dataset. All four approaches were checked to ensure that they all derived the same number of expanded rules and that the sets were identical. For all of our tests this was the case. Thus, the results show that our approach, while reducing the size of the exact basis set does not lose any information and the expanded set of rules can be completely recovered.

6. CONCLUSION & FUTURE WORK

Redundancy in association rules affects the quality of the information presented and this affects and reduces the use of the rule set. The goal of redundancy elimination is to improve the quality and use of the rules, thus allowing them to better solve problems being faced. Our work aims to remove hierarchical redundancy in multi-level datasets, thus reducing the size of the rule set to improve the quality and usefulness, without causing the loss of any information. We have proposed an approach which removes hierarchical redundancy through the use of frequent closed itemsets and generators. This allows it to be added to other approaches which also remove redundant rules, thereby allowing a user to remove as much redundancy as possible.

The next step in our work is to apply this approach to the approximate basis rule set to remove redundancy there. We will also review our work to see if there are other hierarchical redundancies in the basis rule sets that should be removed and will investigate what should and can be done to further improve the quality of multi-level association rules.

REFERENCES

[1] R. Agrawal, T. Imielinski & A. Swami, 'Mining Association Rules between Sets of Items in Large Databases', in ACM SIGMOD International Conference on Management of Data (SIGMOD'93), Washington D.C., USA, 1993, pp 207-216.

[2] R. Agrawal & R. Srikant, 'Fast Algorithms for Mining Association Rules in Large Databases', in 20th International Conference on Very Large Data Bases, Santiago, Chile, 1994, pp 487-499.

[3] F. Bodon, 'A Fast Apriori Implementation', in IEEE ICDM Workshop on Frequent Itemset Mining Implementations (FIMI'03), Melbourne, FL, 2003.

[4] A. Das, W.-K. Ng & Y.-K. Woon, 'Rapid Association Rule Mining', in CIKM'01 ACM, Atlanta, Georgia, USA, 2001.

[5] J. Han & Y. Fu, 'Discovery of Multiple-Level Association Rules from Large Databases', in 21st International Conference on Very Large Databases, Zurich, Switzerland, 1995, pp 420-431.

[6] J. Han & Y. Fu, 'Mining Multiple-Level Association Rules in Large Databases', IEEE Transactions on Knowledge and Data Engineering, Vol 11, pp 798-805, Sep/Oct, 1999.

[7] J. Han & J. Pei, 'Mining Frequent Patterns by Pattern-Growth : Methodology and Implications', ACM SIGKDD Explorations Newsletter, Vol 2, pp 14-20, Dec, 2000.

[8] T.-P. Hong, K.-Y. Lin & B.-C. Chien, 'Mining Fuzzy Multiple-Level Association Rules from Quantitative Data', Applied Intelligence, Vol 18, pp 79-90, Jan, 2003.

[9] M. Kaya & R. Alhajj, 'Mining multi-cross-level fuzzy weighted association rules', in 2nd International IEEE Conference on Intelligent Systems, 2004, pp 225-230.

[10] B. Liu, M. Hu & W. Hsu, 'Pruning and Summarizing the Discovered Associations', in 5th International Conference on Knowledge Discovery and Data Mining (KDD'99), San Diego, CA, 1999, pp 125-134.

[11] B. Liu, M. Hu & W. Hsu, 'Multi-Level Organization and Summarization of the Discovered Rules', in Conference on Knowledge Discovery in Data, Boston, Massachusetts, USA : ACM Press, 2000.

[12] K.-L. Ong, W.-K. Ng & E.-P. Lim, 'Mining Multi-Level Rules with Recurrent Items Using FP'Tree', in 3rd International Conference on Information, Communications and Signal Processing, Singapore, 2001.

[13] N. Pasquier, Y. Bastide, R. Taouil & L. Lakhal, 'Efficient mining of association rules using closed itemset lattices', Information Systems, Vol 24, Issue 1, pp 25-46, 1999.

[14] N. Pasquier, R. Taouil, Y. Bastide, G. Stumme & L. Lakhal, 'Generating a Condensed Representation for Association Rules', Journal of Intelligent Information Systems, Vol 24, pp 29-60, 2005.

[15] Y. G. Sucahyo & R. P. Gopalan, 'CT-ITL : Efficient Frequent Item Set Mining using a Compressed Prefix Tree with Pattern Growth', in 14th Australasian Database Conference, Adelaide, Australia, 2003.

[16] R. S. Thakur, R. C. Jain & K. P. Pardasani, 'Mining Level-Crossing Association Rules from Large Databases', Journal of Computer Science, Vol 12, pp76-81, 2006.

[17] Y. Xu & Y. Li, 'Mining Non-Redundant Association Rules Based on Concise Bases', International Journal of Pattern Recognition and Artificial Intelligence, Vol 21, pp 659-675, Jun, 2007.

[18] Y. Xu, & Y. Li, 'Generating Concise Association Rules', in 16th ACM Conference on Conference on Information and Knowledge Management (CIKM'07), Lisbon, Portugal, 2007, pp 781-790.

[19] M. J. Zaki, 'Generating Non-Redundant Association Rules', in Proceedings of the KDD Conference, 2000, pp 34-43.

[20] M. J. Zaki, 'Mining Non-Redundant Association Rules', Data Mining and Knowledge Discovery, Vol 9, pp 223-248, 2004.